

## Лабораторная работа №6

### ПРОГРАММИРОВАНИЕ С ИСПОЛЬЗОВАНИЕМ СТРОК

#### *Индивидуальные задания*

(Данные задания выполнять в соответствии с выданным вариантом, оформлять отчет, который должен включать задание, блок-схему и текст программы) Рассматривать строки как одномерный массив символов.

#### *Задание 1.*

При выполнении задания предусмотреть возможность ввода текста с клавиатуры. При работе с текстом использовать строки длиной не более 80-ти символов. Исходный текст и полученный результат вывести на экран.

1. В строке, состоящей из групп нулей и единиц, разделенных пробелами, найти количество групп с пятью цифрами.
2. В строке, состоящей из групп нулей и единиц, найти и вывести на экран самую короткую группу.
3. В строке, состоящей из групп нулей и единиц, подсчитать количество символов в самой длинной группе.
4. В строке, состоящей из групп нулей и единиц, найти и вывести на экран группы с четным количеством символов.
5. В строке, состоящей из групп нулей и единиц, подсчитать количество единиц в группах с нечетным количеством символов.
6. Из строки, состоящей из букв, цифр, запятых, точек, знаков + и – , выделить подстроку, которая соответствует записи целого числа.
7. Из строки, состоящей из букв, цифр, запятых, точек, знаков + и – , выделить подстроку, задающую вещественное число с фиксированной точкой.
8. Из строки, состоящей из букв, цифр, запятых, точек, знаков + и – , выделить подстроку, задающую вещественное число с плавающей точкой.
9. Дана строка символов, состоящая из цифр, разделенных пробелами. Вывести на экран числа этой строки в порядке возрастания их значений.
10. Дана строка символов, состоящая из цифр, разделенных пробелами. Вывести четные числа этой строки.
11. Дана строка, состоящая из слов, разделенных пробелами. Вывести на экран порядковый номер слова, содержащего  $k$ -ю позицию, если в  $k$ -й позиции пробел, то – номер предыдущего слова. Значение  $k$  ввести с клавиатуры.
12. Дана строка, состоящая из слов, разделенных пробелами. Разбить исходную строку на две подстроки, причем первая длиной  $k$  символов, если на  $k$ -ю позицию попадает слово, то его следует отнести ко второй строке. Значение  $k$  вводится с клавиатуры.

13. Дана строка, состоящая из слов, разделенных пробелами. Вывести на экран порядковый номер слова максимальной длины и номер позиции строки с которой оно начинается.

14. Дана строка, состоящая из слов, разделенных пробелами. Вывести на экран порядковый номер слова минимальной длины и количество символов в этом слове.

15. В строке символов посчитать количество скобок различного вида.

### **Задание 2.**

1. Напечатать квитанцию об оплате за телеграмму, если известна стоимость одного слова.

2. В тексте слова заданной длины заменить указанной подстрокой, длина которой может не совпадать с длиной слова.

3. В тексте каждую букву заменить ее номером в алфавите.

4. В тексте после указанного символа вставить подстроку.

5. После каждого слова текста, оканчивающегося заданной подстрокой, вставить указанный символ.

6. В тексте удалить указанный символ везде, где он встречается.

7. Из текста удалить все символы, не являющиеся буквами, кроме пробелов.

8. Из текста удалить все слова заданной длины, начинающиеся с согласных букв.

9. Найти, каких букв в тексте больше – гласных или согласных.

10. В тексте найти и вывести на экран все слова максимальной и минимальной длины.

11. В тексте найти и напечатать слова, начинающиеся и оканчивающиеся гласной буквой.

12. В тексте найти и напечатать символы, встречающиеся наиболее часто.

13. В тексте нет слов, начинающихся одинаковыми буквами. Напечатать слова текста в таком порядке, чтобы последняя буква каждого слова совпадала с первой буквой последующего слова. Если все слова нельзя напечатать в таком порядке, найти такую цепочку, состоящую из наибольшего количества слов.

14. Найти наибольшее количество предложений текста, в которых есть одинаковые слова.

15. Напечатать без повторения слова текста, у которых первая и последняя буквы совпадают.

### **Задание 3. Стандартных функций работы со строками не использовать.**

1. Ввести матрицу. Элементы матрицы – строки. Подсчитать сумму кодов символов каждого слова и, если сумма оказалась четной, развернуть зеркально это слово в строке. Полученные матрицы вывести на экран. Стандартных функций работы со строками не использовать.

2. Обработать массив строк (длина каждой строки не более 100

символов). Если в строке встречена последовательность одинаковых символов, заменить их кодом 255, за которым следует код этого символа и количество одинаковых символов. Стандартных функций работы со строками не использовать.

3. Написать программу, которая, анализируя массив, состоящий из строк, выводит на экран количество слов женского, мужского и среднего рода. Стандартных функций работы со строками не использовать.

4. Имеется массив строк. На том же месте, не заводя других массивов, записать слова в обратном порядке, рассматривая все строки, т.е. первое слово первой строки станет последним словом последней строки (если объединить строки, будет гораздо проще). Стандартных функций работы со строками не использовать.

5. В массиве строк найти среднее значение длины строки. Строки, длина которых больше среднего, – обрезать, меньше – добавить пробелы. Полученный массив вывести на экран. Стандартных функций работы со строками не использовать.

6. Ввести строку символов, содержащую два или более слов, разделенных пробелами. Написать функцию, меняющую местами все четные и нечетные слова в строке, предполагая, что за один раз можно менять местами не более двух символов. Результат вывести на экран. Стандартных функций работы со строками не использовать.

7. Написать функции (копирование, сравнение, объединение) для работы со строками, используя указатели. Стандартных функций работы со строками не использовать.

8. Используя указатели, написать функции перевода целых чисел в строку и строки в число; функции перевода вещественных чисел в строку, перевода строки в вещественное число. Стандартных функций работы со строками не использовать.

9. Ввести массив строк. В функции для каждой строки проверить, является она симметричной или нет. (Симметричной считается строка, которая одинаково читается слева направо и справа налево). Вывести на экран саму строку и результат её обработки. Стандартных функций работы со строками не использовать.

10. Ввести массив строк. В функции для каждой строки проверить, является она палиндромом (симметричной с точностью до пробелов) или нет. Например, палиндромами являются цепочки: «АРГЕНТИНА МАНИТ НЕГРА» и «А РОЗА УПАЛА НА ЛАПУ АЗОРА».

Результат вывести на экран. Стандартных функций работы со строками не использовать.

11. Ввести строку и вычислить сумму входящих в неё цифр, причем знак очередного слагаемого должен быть противоположным знаку предыдущего слагаемого. Например: для строки «asdd1vnb24vnf63vbn,-5h-2kk» сумма  $S = 1 - 2 + 4 - 6 + 3 - 5 + 2 = -3$ . Результат вывести на экран. Стандартных функций работы со строками не использовать.

12. Ввести строку и определить наибольшее записанное в этой строке

целую часть числа (без учета знака числа). Например, для строки «sdfvgsd1.9fdmjgvb15.25dnj05» наибольшее целое число 15. Результат вывести на экран. Стандартных функций работы со строками не использовать.

13. Ввести строку и определить все входящие в неё символы. Например: строка «abscbbbabba» состоит из символов «a», «b» и «c». Результат вывести на экран. Стандартных функций работы со строками не использовать.

14. Ввести строки символов. Среди литер этого текста особую роль играет знак #, появление которого означает отмену предыдущей литеры текста; k знаков # отменяют k предыдущих литер (если такие есть). Преобразовать строку с учетом роли знака #. Например, строка «VR##Y#HELO#LO» должна быть напечатана в виде: «HELLO». Результирующую строку вывести на экран. Стандартных функций работы со строками не использовать.

15. Ввести две строки. В функции определить, являются ли они анаграммами (т.е. одна строка получена из другой перестановкой букв). Например: строки БУК и КУБ или СОЛЬ и ЛОСЬ являются анаграммами. Стандартных функций работы со строками не использовать.

## **Дополнительно\***

### **Задание 1.**

1 апреля 20?? года будет проведена реформа английского языка, позволяющая облегчить его освоение иностранцами и английскими школьниками. Во-первых, из алфавита уберут четыре буквы **C, Q, W** и **X** (на клавиатуре компьютера вместо них будут клавиши, вызывающие программы **Word**, **eXcel** и другие). Вместо **c** перед буквами **e, i, y** нужно будет писать букву **s**, а в остальных случаях – букву **k**. Вместо буквы **q** нужно будет писать букву **k**, вместо сочетания **qu** – **kv**, вместо **x** – **ks**, а вместо **w** – **v**. Во-вторых, сочетание букв **ph** будет записываться как **f**, **you** и **oo** – как **u**, **ee** – как **i**, **th** – как **z**. Кроме того, все двойные согласные (включая образовавшиеся после замен), вызывающие большие трудности у учащихся, станут одинарными, например, **apple** после реформы нужно писать как **aple**.

В связи с реформой нужно переработать огромное количество текстов. Напишите программу, выполняющую эту работу.

### **Примечание для любознательных**

Основой для задачи послужило несколько юмористических статей:

The European Union commissioners have announced that agreement has been reached to adopt English as the preferred language for European communications, rather than German, which was the other possibility. As part of the negotiations, Her Majesty's Government conceded that English spelling had some room for improvement and has accepted a five-year phased plan for what will be known as EuroEnglish (Euro for short).

In the first year, "s" will be used instead of the soft "c". Certainly sivil servants will resieve this news with joy. Also, the hard "c" will be replased with "k". Not only will this klear up konfusion, but typewriters kan have one less letter.

There will be growing publik enthusiasm in the sekond year, when the troublesome "ph" will be replaced by "f". This will make words like "fotograf" 20 persent shorter.

In the third year, publik akseptanse of the new spelling kan be expekted to reach the stage where more komplikated changes are possible. Governments will enkorage the removal of double letters, which have always ben a deterrent to akurate speling. Also, al will agre that the horrible mes of silent "e"s in the languag is disgrasful, and they would go.

By the fourth year, people wil be reseptiv to steps such as replasing "th" by "z" and "w" by "v". During ze fifz year, ze unesesary "o" kan be dropd from vords kontaining "ou" and similar changes vud of kors be aplid to ozer kombinations of leters.

After ze fifz yer, ve vil hav a reli sensibl riten styl. Zer vil be no mor trubls or difikultis and evrivun vil find it ezi tu understand ech ozer.

Ze drem vil finali kum tru.

### **A Plan for the Improvement of English Spelling by M. J. Shields (Mark Twain?)**

For example, in Year 1 that useless letter "c" would be dropped to be replased either by "k" or "s," and likewise "x" would no longer be part of the alphabet. The only kase in which "c" would be retained would be the "ch" formation, which will be dealt with later. Year 2 might reform "w" spelling, so that "which" and "one" would take the same konsonant, wile Year 3 might well abolish "y" replasing it with "i" and lear 4 might fiks the "g / j" anomali wonse and for all.

Jenerally, then, the improvement would kontinue iear bai iear with lear 5 doing awai with useless double konsonants, and lears 6-12 or so modifaiing vowlz and the rimeining voist and unvoist konsonants. Bai lear 15 or sou, it wud fainali bi posibl tu meik ius ov thi ridandant letez "c," "y," and "x" – bai now jast a memori in the maindz ov ould doderez – tu riplais "ch," "sh," and "th" rispektivli.

Fainali, xen, aafte sam 20 iers ov orxogrefkl riform, wi wud hev a lojikl, kohirnt speling in ius xrewawt xe Ingliy-spiking werld.

### **Входные данные**

Во входном файле содержится текст на английском языке, без переносов слов. Длина строк не превышает **100** символов.

### **Выходные данные**

В выходной файл вывести тот же текст, но уже соответствующий реформе. Если первая буква заменяемого сочетания букв была прописной, то первая буква замены должна быть также прописной. Вторая буква в заменах **x**→**ks**, **qu**→**kv** должна быть всегда строчной.

**Входные данные**

Too swift for Theex, too quick for Gallow,  
Too strong for young Prince Joseph to follow.

**Выходные данные**

Tu svift for Ziks, tu kvik for Galov,  
Tu strong for ung Prinse Josef to folov.

***Задание 2.***

Во многих прикладных задачах, таких как поиск в сети или расшифровка генома, требуется совершать некоторые действия со строками. Например, часто требуется по некоторым данным о строке восстановить ее саму.

Вам даны две строки **S<sub>1</sub>** и **S<sub>2</sub>**. Известно, что одна из них была суффиксом искомой строки **S**, а другая — ее префиксом. Также известна длина искомой строки **L**, а также то, что строка **S** состояла только из строчных букв латинского алфавита.

Необходимо определить число строк, удовлетворяющих этим ограничениям. Так как это число может быть достаточно большим, то необходимо вывести его по модулю **m**.

**Входные данные**

Первая строка содержит одно целое число **t** ( $1 \leq t \leq 100$ ) — количество наборов входных данных, которые необходимо обработать.

Описание каждого из наборов входных данных содержит три строки. Первая из них содержит два целых числа: **L** и **m** ( $1 \leq L \leq 10^9$ ,  $1 \leq m \leq 10^4$ ). Вторая и третья строки содержат соответственно строки **S<sub>1</sub>** и **S<sub>2</sub>**. Они непусты, состоят из строчных букв латинского алфавита, их длины не превышают **200** символов.

**Выходные данные**

Для каждого набора входных данных выведите в отдельной строке остаток от деления числа строк, удовлетворяющих требованиям условия, на **m**.

| Входные данные                | Выходные данные |
|-------------------------------|-----------------|
| 3<br>14 1000<br>cup<br>russia | 752<br>0<br>1   |

|   |  |
|---|--|
| 7 123<br>russian<br>codecup<br>7 15<br>codec<br>decup |  |
|---|--|

### ***Задание 3.***

Анаграммой слова называется любая перестановка всех букв слова. Например, из слова **SOLO** можно получить **12** анаграмм: **SOLO, LOSO, OSLO, OLSO, OSOL, OLOS, SLOO, LSOO, OOLS, OOSL, LOOS, SOOL**.

Напишите программу, которая выводит количество различных анаграмм, которые могут получиться из этого слова.

#### **Входные данные**

Слово, количество букв в котором не превышает **14**.

#### **Выходные данные**

Количество различных анаграмм.

#### **Входные данные #1**

SOLO

#### **Выходные данные #1**

12